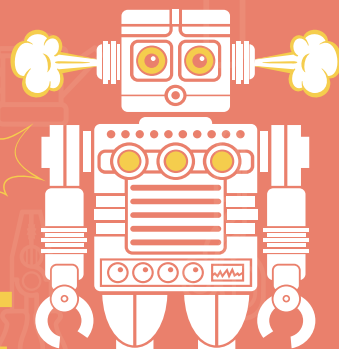
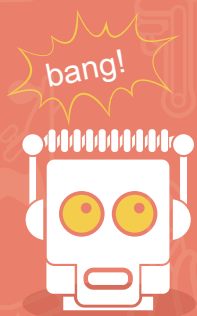
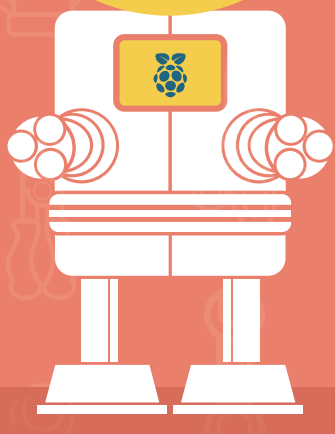
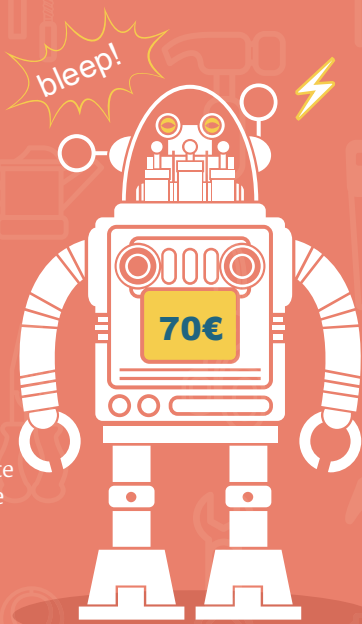
COSTRUIRE RASPBERRY PI ROBOT PER 70€

Cercavi un piccolo progetto per favorire una rivolta dei robot? Costruiamo il nostro piccolo automa e conquistiamo il mondo O il tuo soggiorno ...

DAI UN NOME AL NOSTRO ROBOT!

Non abbiamo ancora trovato un nome al nostro Raspberry Pi Robot.Twitta i tuoi suggerimenti @TheMagPi e sceglieremo il migliore

Se non avete mai costruito un robot, non potete sapere veramente cosa vi state perdendo. Il processo di assemblaggio e arrivare a farlo funzionare è esattamente come si vede nei film. L'euforia e l'orgoglio nel vedere il vostro prodotto finale che trotterella in giro, è ineguagliabile .

Anche se ci sono sul mercato diversi ottimi kit facili da montare, ci piaceva l'idea di realizzare un altro manuale: creare un robot fai-da-te. inoltre non vogliamo che tu dobbiate fare rapine in banca o impegnarvi in settimane di lavoro. Abbiamo quindi deciso di sederci intorno a un tavolo e pensare a un robot che sia realizzabile in un solo weekend, con una spesa di 70€ incluso Il Raspberry Pi. Quindi vestitevi, mettete gli stivali e inforcate la moto per andare a cercare tutto quello di cui avrete bisogno per costruire il vostro piccolo amico robot personale.

LA LISTA DELLA SPESA

Quel che ti serve per costruire
Il tuo robot a buon mercato

[1] Raspberry Pi A+

Nonostante sia la scheda piu' piccola ed economica della famiglia RaspberryPi, la A+ e' sicuramente abbastanza potente per elaborare tutto quello che serve per fare diventare questo robot una realtà . Inoltre lo useremo anche come telaio, e' abbastanza robusta per sostenere tutto quello che abbiamo in mente.

21.50€

[2] Micro motoriduttori

Per il nostro piccolo robot realizzato su un A+ , abbiamo bisogno di un paio di motori abbastanza piccoli da connettere alla scheda , e abbastanza potenti per muoverlo. Questi "ragazzi" sono quello che fanno al caso nostro.

3.80€ l'uno

bit.ly/1g7Tva0

[3] Ruote

Avremo bisogno delle ruote per i motori. Indovinate perché...
Queste sono sufficientemente piccole e perfette per noi.

5.50€ la coppia

bit.ly/1g7UaZe

[4] Rotelle a sfera

Abbiamo scelto di usare delle ruote a sfera sul retro del nostro robot invece di utilizzare altri motori. Così facendo, si mantiene un prezzo basso e ne guadagna la semplicità nelle manovre. Ne sarebbe in realtà sufficiente anche solo una, ma ne abbiamo messe due nella lista per averne una di scorta da utilizzare come ricambio.

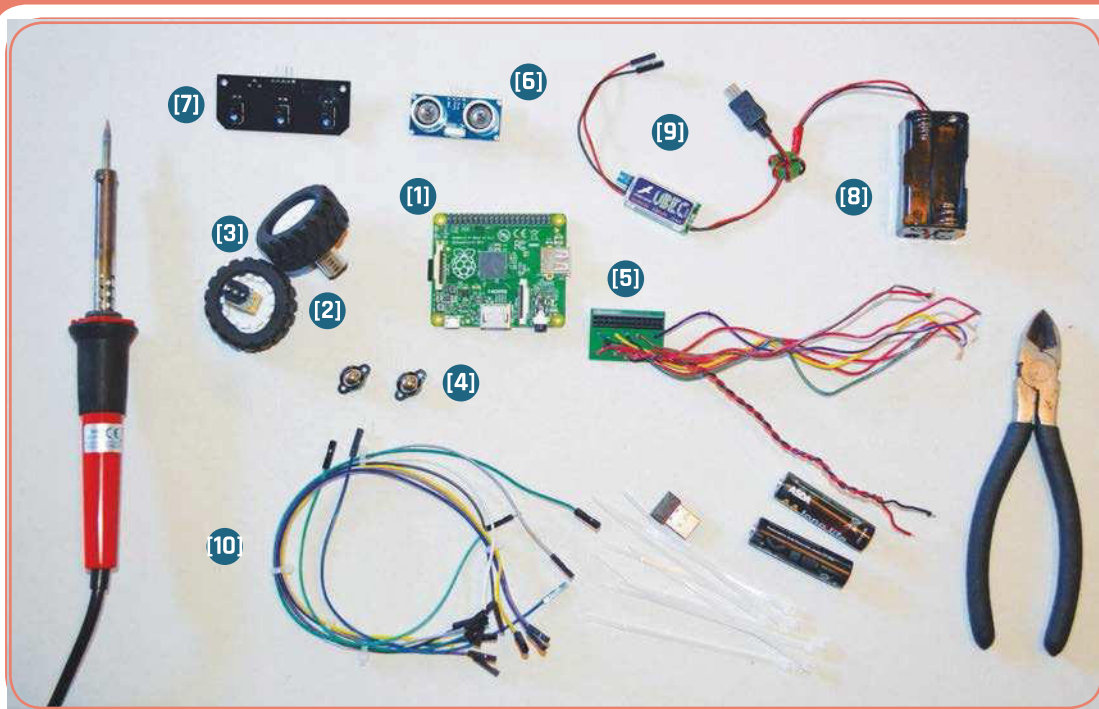
2.80€

bit.ly/1XHPIHX

[5] Scheda motori PicoBorg

La piccola scheda controllo motori PicoBorg e' in grado di controllare quattro motori distinti utilizzando esclusivamente un Raspberry Pi e una modesta alimentazione esterna. Ha anche una dimensione perfetta per il nostro piccolo robot.

9.70€

piborg.org/picoborg

[6] Sensore ultrasuoni HC-SR04

Non vogliamo certo che il nostro robot finisca dritto contro il muro, montando un sensore ad ultrasuoni sarà in grado di sapere quando è il caso di fermarsi... oppure chi inseguire.

3.50€

bit.ly/1JSxo29

[7] Inseguitore linee a tre vie

E' come mettere delle piccole ali al nostro robot, I sensori a inseguimento di linee Ryanteck sono perfetti per i nostri scopi, e hanno anche un prezzo molto abbordabile.

7€

bit.ly/1NgC4kS

[8] Contenitore batterie

Servirà a contenere le batterie che alimenteranno il nostro robot..
È possibile prenderne uno qualunque per pochi spiccioli

1€

[9] Cavo batteria UBEC

A meno che non troviate divertente avere il vostro robot al guinzaglio tramite il cavo di alimentazione, Avrete bisogno uno di questi incredibili cavi della Alba Robotics per convertire l'ingresso della batteria in un segnale che non danneggi il Raspberry Pi.

8.30€

bit.ly/1HLKih7

[10] Minuterie varie

Avrete bisogno di alcuni di cavi, Blu-Tack, batterie, un paio di resistenze, un dongle wireless, e un saldatore per collegare tutto in maniera stabile.
Se non avete questi materiali, questo è un buon momento per procurarseli.

TOTALE: 66.90€

PREPARAZIONE COMPONENTI

Saldature, calbaggi e altro ancora...

UN PO' DI ISPIRAZIONE

Utilizzare un Raspberry Pi Modello A+ come telaio? Motori cablati? Se hai seguito con molta attenzione ogni cosa pubblicata sul blog ufficiale di Raspberry Pi, questo potrebbe suonarti un po' famigliare e notare più di una somiglianza con il Toast-E robot realizzato da Gert van Loo lo scorso anno. Abbiamo preso una piccola ispirazione dal suo progetto per costruire il nostro robot, che funziona similarmente al suo. Leggi qualcosa di più in merito qui: **gertbot.com/toaste.html**

Prima di iniziare la costruzione del robot, bisogna preparare i componenti e il Raspberry Pi da utilizzare per questo particolare progetto. Cominciamo con il Raspberry Pi.

Preparare Raspberry Pi

Se non ne avete una già pronta, create una scheda SD per il Modello A+. Eseguite gli aggiornamenti necessari per avere le ultime versioni rilasciate per programmi e kernel, per il momento il log in sul desktop. Se non lo è già, collegate il dongle WiFi. E' possibile controllare il robot via SSH al volo tramite collegamento alla rete wireless, avendo così un migliore controllo.

Effettuate la configurazione della connessione WiFi semplicemente utilizzando l'interfaccia grafica e il Pi ricorderà la password e la rete wireless.

Usando il DHCP, il router probabilmente assegnerà al Pi lo stesso indirizzo IP quando si connetterà; per assicurarsi però che questo avvenga ogni volta, è preferibile impostare un indirizzo IP statico.

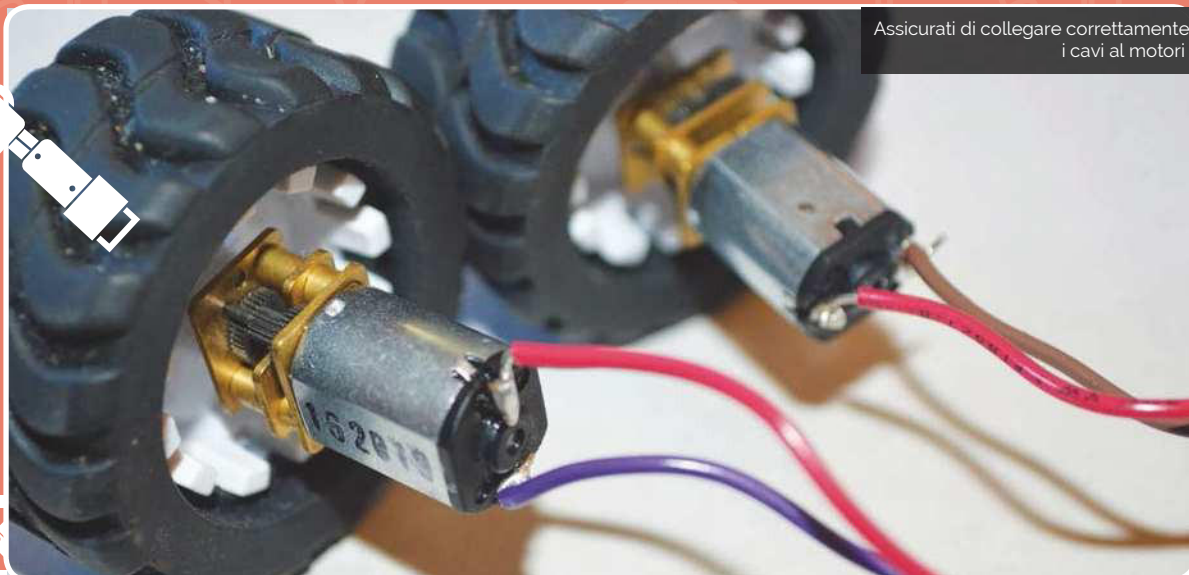
Aprite il terminale e usate il comando **ifconfig** per determinare l'indirizzo IP che il DHCP ha assegnato al Raspberry (inet addr), il Broadcast Range (Bcast) e la Subnet Mask (Mask). Una volta annotati questi dati, utilizzate **sudo route -n** per capire la Destination e l'indirizzo del Gateway per conoscere la rete e il router.

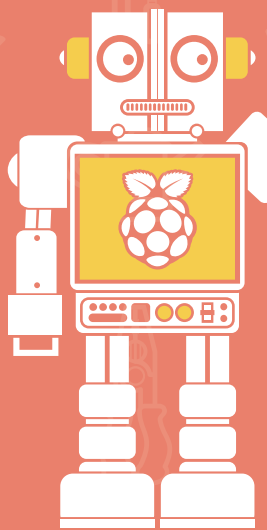
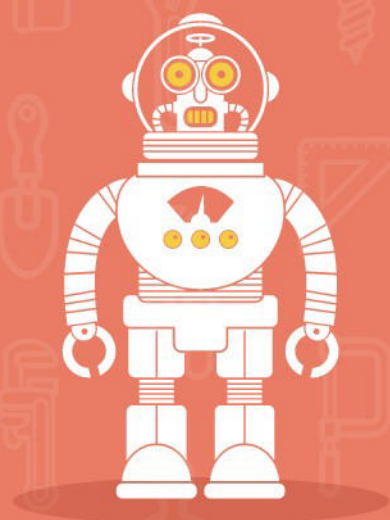
Aprire i file delle interfacce di rete (da terminale, digitate **sudo nano /etc/network/interfaces**) per poter visualizzare le impostazioni correnti dei dispositivi di rete. Le vostre impostazioni wireless dovrebbero essere qualcosa di simile a questo:

```
allow-hotplug wlan0
iface wlan0 inet manual
wpa_roan /etc/wpa_supplicant/
wpa_supplicant.conf
```

Cambiate **manual** con **static** e aggiungete le seguenti 5 linee:

```
address [l'indirizzo IP che
volete utilizzare]
netmask [indirizzo netmask]
network [la destination]
broadcast [broadcast range]
gateway [indirizzo del
gateway]
```

Salvate e dopo il riavvio l'IP Sarà statico. Se lo desiderate, è possibile anche lanciare **sudo raspi-config** e lì impostare il boot in modalita' desktop prima di proseguire con il resto del robot.

Assicurati di collegare correttamente i cavi al motori

Il cablaggio della scheda motori è abbastanza semplice, ma seguite i nostri consigli per non sbagliare.

Saldare la PicoBorg

Nell' immagine, si può vedere che abbiamo molti fili extra che non sono usati sulla nostra PicoBorg: questo perché troviamo predisposto il tutto per quattro motori. Per questo progetto, avete bisogno di fare le saldature solo per il primo e per il secondo motore, oltre il cavo delle batterie che pilotano i motori. Assicuratevi di avere abbastanza cavi per collegare i diversi ingressi e uscite, e vi suggeriamo di utilizzare del filo rosso per il positivo e nero per il negativo.

Per la PicoBorg, è consigliabile utilizzare filo unipolare di buona qualità e fare attenzione, durante la saldatura, a non surriscaldare eccessivamente la scheda PCB. Potete trovare diversi suggerimenti su come saldare la PicoBorg sulla sua pagina (**piborg.org/picoborg/specs**), oppure spendere qualcosina in più per acquistarne una già pre-saldata.

Una volta connesso, dovrai saldare una coppia di fili sul lato superiore della scheda: abbiamo bisogno sia dei 5V che dei 3,3V rispettivamente per il sensore ad ultrasuoni e per l'inseguitore di linee, e non ne resterà nessuno disponibile dei rimanenti 14 pin del GPIO una volta che il PicoBorg sarà posizionato al suo posto.. Dall'immagine, si può vedere che abbiamo collegato un filo con plug femmina al pin 1 e al pin 2 del Connettore: quando innesteremo la scheda sul Raspberry Pi, questi sono i due pin più vicini al bordo e alla SD card.
Il pin all'angolo e' il pin 2, che porta la tensione positiva di 5V .
Il pin successivo a questo e' il 3V3 pin (tensione positiva di 3,3V).

Saldare i motori

I motori non hanno i fili già collegati, dovete quindi solo saldarci su direttamente i fili provenienti dalla PicoBorg.
È una operazione tutt'altro che complicata: l'unica cosa a cui prestare attenzione è rispettare Il polo positivo e quello negativo per collegare i motori con la guista polarità.

Collegare portabatterie

Abbiamo bisogno delle batterie per alimentare sia la scheda con L' UBEC, sia i motori. Quello che abbiamo fatto è spelare i fili di pochi mm, e usare i fili della scheda motori saldandoli direttamente al contenitore delle batterie.
L' UBEC è connesso, invece, tramite il suo connettore femmina sull'estremità dei fili. Finito il tutto ti consigliamo di dare una passata di nastro isolante per evitare ogni possibile contatto Indesiderato!

Collegare le resistenze al sensore a ultrasuoni

Il sensore a ultrasuoni dovrà principalmente connettersi ai pin GPIO, con l'eccezione del pin ECHO sulla HC-SR04 che richiede alcune resistenze per lavorare correttamente: resistenze da 1K e 2K. La resistenza da 1K andrà inserita tra il pin della GPIO e la quello denominato porta ECHO. Quella da 2K, invece,collegata il pin della GPIO e il pin di massa che stiamo usando. Come si vede nell' immagine, saranno molto vicini l'uno all'altro, cosa che, tra l'altro, permetterà di effettuare questi collegamenti senza la necessità di utilizzare breadbord o altre schede.

ALTERNATIVE

Voglia di spendere qualche soldo in più?

Ecco cosa potresti fare..

TELAIO DAGU 2WD

bit.ly/1Njk1dP

Ha dato l'ispirazione al nostro robot per quanto riguarda la configurazione delle ruote , questo telaio venduto dalla Dawn Robotics è uno dei più semplici in circolazione. è decisamente piu' grande dell' A+, e quindi e' possibile montarci sopra anche un Raspberry Pi B+ oppure 2 insieme per avere più potenza possibile. E' abbastanza economico e può essere una buona scelta in caso di evoluzione.

SENSE HAT

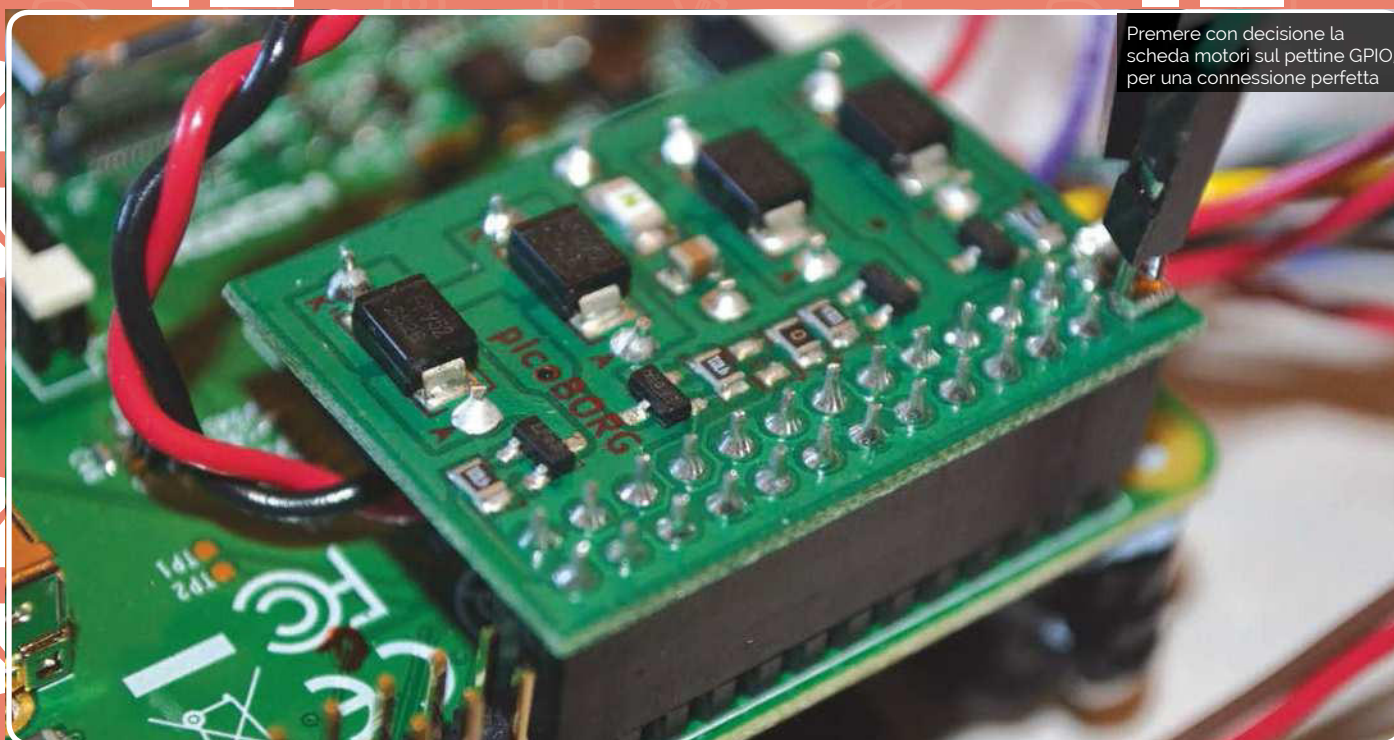
La Sense Hat è in grado di rilevare grandezze extra, ad esempio se si sta andando in salita o quanto veloce ci si sta muovendo. Può anche visualizzare Piccole immagini, come frecce per mostrare la direzione di spostamento. Potrete dare al vostro robot più vitalità e renderlo più interessante, senza usare tutti i piedini della GPIO.

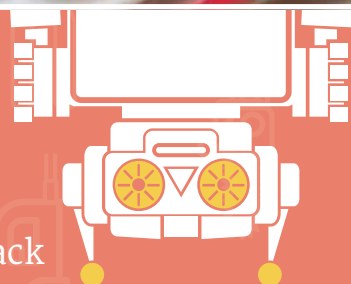
BUDGET ROBOTICS KIT

bit.ly/1L3Jmbw

Volete qualcosa che sia più semplice da assemblare ma con le stesse capacità d'espansione? Il kit Ryanteck di Budget Robotics, richiede che siate già In possesso di un Raspbberry Pi, mentre tutto il resto, compresi motori e alimentazione, è compreso nel kit.

Telaio Dagu 2WD

Premere con decisione la scheda motori sul pettine GPIO, per una connessione perfetta

CONSTRUISCI IL TUO ROBOT

Colleghiamo tutto assieme. Non scordare il Blu-Tack

La prima cosa da fare è connettere il Raspberry Pi alla scheda motori. La maggior parte dei componenti principali si collega ad essa in vari modi, quindi è preferibile iniziare con questa scheda prima di incasinarci con tutti gli altri cavi. Utilizzate la nostra immagine per capire il corretto orientamento quando inserite la PicoBorg sul GPIO: dovrà essere a filo della scheda del Raspberry Pi invece di sporgere all'esterno; notate che si inserisce sui pin dal lato opposto rispetto alla porta USB.

Montate le ruote sui motori e dopo fissateli: abbiamo utilizzato una fascetta da elettricista passandola attraverso i fori delle viti di fissaggio per bloccare saldamente i motori sul lato della PCB. Accertatevi che siano fissati nello stesso modo e che tuttavia rimanga spazio sufficiente alle ruote per girare senza toccare sul lato della scheda del Pi.

Fissate la ruota a sfera sulla parte posteriore del Pi: ne abbiamo messa una singola sullo slot della scheda microSD, ma volendo, voi potete metterne due, una per lato. Questo tipo di rotelle hanno solitamente a corredo dei distanziali, che possono essere usati per tenete il Pi sollevato dal piano.
Noi abbiamo usato il Blue-Tak per attaccarla, ma è meglio usare un paio di fascette sottili!

Abbiamo usato il Blue-Tak anche per attaccare il sensore a ultrasuoni e l'inseguitore di linee: quest' ultimo appiccicato sotto al PCB, mente il sensore a ultrasuoni sulle porte USB. A dire il vero, così non è molto stabile, quindi, se trovate un altro sistema di fissaggio, attuatelo pure.

Ora dovrete collegare tutto assieme. I fili di alimentazione saldati alla PicoBorg devono essere collegati ai terminali corretti (il sensore a ultrasuoni richiede 5V, che è il pin 2; mentre l'altro va all' inseguitore di linee), fate quindi riferimento allo schema (nella prossima pagina) per verificare dove vanno collegati gli altri fili. Fortunatamente, c'è un bel po' di spazio libero sul connettore GPIO.

L'ultima cosa che resta da fare è mettere il pacco batterie sul Raspberry Pi: dovrebbe essere rimasto dello spazio accanto alla PicoBorg. Ora penserete che il passo successivo sia di collegare il cavo UBEC e avviare il Pi, ma non non è ancora arrivato quel momento.
E' meglio programmare correttamente il robot, prima, in modo che possiate usare i motori esattamente nel modo che desiderate e in piena sicurezza, prima di dare alimentazione alll'intero robot.

SMONTAGGIO

Vi siete divertiti per un po' con il robot e ora vorreste avere indietro alcuni pezzi, per riutilizzarli in altri progetti? La maggior parte del robot è facile da smontare - basta rimuovere pochi fili – ma dovrete anche a tagliare quelle fascette da elettricista e probabilmente dissaldare il pacco batterie e fili aggiuntivi sulla scheda dei motori per far si che sia tutto utilizzabile altrove . Se trovate un metodo più semplice e pratico di costruire il robot, che renda più facile smontare tutto per riutilizzare le sue parti, non esitate a metterlo in pratica!

PROGRAMMA IL TUO ROBOT

I passi finali prima di sguinzagliare
Il vostro robot in giro per il mondo

>PASSO-01

Primo avvio

Collegare una fonte di alimentazione esterna al Raspberry Pi (non il pacco batterie e cavi UBEC peril momento) e effettuate l'avvio.
Attendete circa 30 secondi o giù di lì per provare l'accesso - di solito si conta da quando il dongle Wi-Fi inizia a lampeggiare – e poi connettersi al Pi utilizzando SSH da un altro computer. Potete utilizzare il terminale di OS X o di Linux , o usando PuTTY su Windows .
inserite l'indirizzo IP statico che avete impostato nella connessione, con utente **pi** e password **raspberry**.

>PASSO-02

Installare la PicoBorg

La PicoBorg è la cosa principale da installare per far funzionare il il robot. Per farlo, occorre prima creare una nuova directory con nome picoborg digitando **mkdir picoborg**, entrare in essa con **cd picoborg**, e successivamente , scrivere i seguenti comandi:

```
$wget http://www.piborg.
org/downloads/picoborg/
examples.zip
$unzip examples.zip
$chmod +x install.sh
$./install.sh
```

>PASSO-03

Scriviere il codice

Usare cd per tornare alla directory home, con **cd ~**, e creare una nuova directory per il nostro robot. Chiamiamola robotcode (**mkdir robotcode**) e usiamo nuovamente cd per entrarci. Ora possiamo scrivere il listato per l'inseguimento di linee che consentirà al Raspberry Pi siseguire automaticamente un percorso e diventare quindi un vero robot autonomo. Digitate:

```
$ nano MagPibot.py
```

... E copiare il codice da l'esempio di codice sulla pagina, o scaricarlo dal nostro sito web e copiare e incollare il codice nel terminale. Salvarlo.

>PASSO-04

Primi test

Se le batterie sono già inserite nel contenitore, toglietele. Serve per essere certi che i motori non possano partire non appena cominciamo il test.
Per eseguire lo script, dovrete digitare:

```
$ sudo python MagPibot.py
```

È necessario eseguirlo come sudo per accedere ai pin GPIO correttamente .
Inizializzerà a livello alto tutti i pin GPIO e controllerà che tutto sia attivo prima di chiedere di premere il tasto INVIO per continuare. Se i motori fossero stati alimentati, non sareste stati in grado di lanciare da qui il programma completo. Per ora, uscire dal programma con CTRL+C .

>PASSO-05

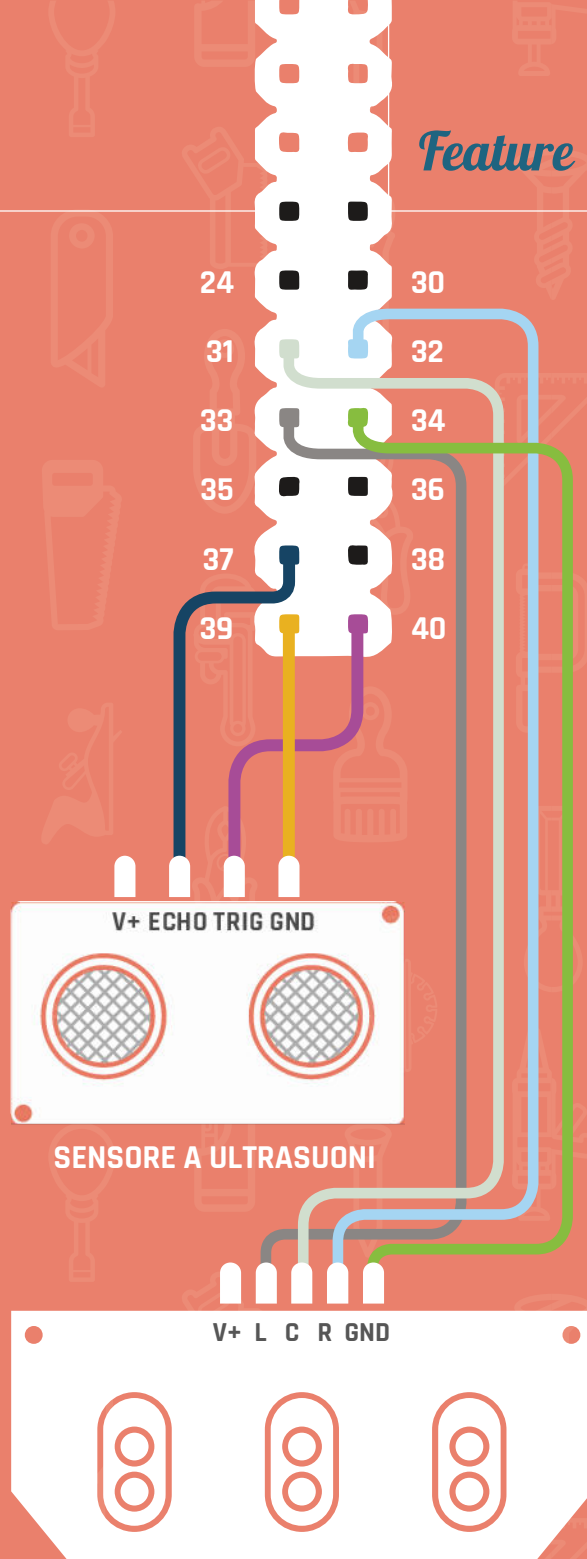
Prima corsa

Fate uno shutdown del Raspberry Pi e rimettete le batterie nel loro portabatterie. Togliete anche l'alimentazione esterna e collegatela nell' UBEC. Il Raspberry Pi dovrebbe riavviarsi come prima, e poi potrete collegarvi ad esso tramite SSH. Se non avete già predisposto un tracciato con una linea da seguire, non c'è problema: per ora potete farlo semplicemente correre su una superficie. Eseguite lo script Python come nel passo precedente, e fare in modo di stare sempre vicino al Raspberry Pi per afferrarlo nel caso in cui cominci a correre un po' troppo in fretta.
Ricordate , è possibile spegnerlo con CTRL + C .

24 30
31 32
33 34
35 36
37 38
39 40

V+ ECHO TRIG GND

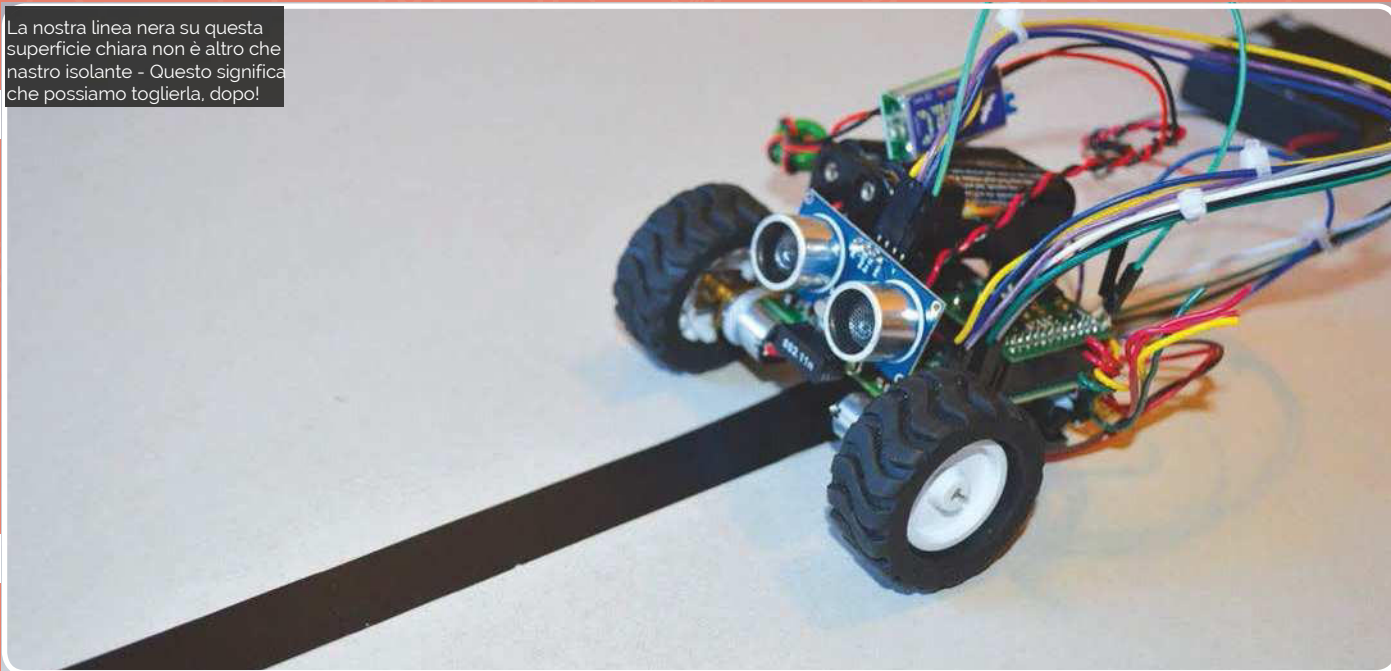
SENSORE A ULTRASUONI

V+ L C R GND

INSEGUITORE DI LINEE A 3 VIE

>PASSO-06

Modifiche e aggornamenti

Il nostro codice è servito ai nostri scopi , ma potrebbe non essere esattamente quello di cui il vostro robot ha bisogno. Nella pagina seguente, spiegheremo le parti salienti del codice in modo che possiate modificarlo a vostro piacimento; questo sopratutto in merito a come il robot può reagire alla linea seguita o a qualsiasi oggetto trovi sul suo cammino. È possibile aprire lo script Python come abbiam fatto in precedenza con **nano MagPibot.py** e modificarlo da lì.

La nostra linea nera su questa superficie chiara non è altro che nastro isolante - Questo significa che possiamo toglierla, dopo!

FUNZIONI BASE DI UN ROBOT

Come controllare il vostro robot e come può controllarsi da solo

Seguire una linea

Il listato di codice nella pagina successiva ha una funzione molto specifica che permette al robot che abbiamo costruito di seguire una linea. Qui si vede il nostro robot che prova a seguire del nastro nero fino a destinazione. Il sensore che abbiamo attaccato sul fondo del Pi può rilevare se c'è una linea nera , oppure se c'è una superficie bianca o chiara . Ritorna un valore basso quando è sopra la linea nera (0) , e lo utilizzeremo per insegnare al nostro robot a controllare se stesso. Essendo interessati solo a correzioni di direzione, utilizzeremo solo i sensori di destra e di sinistra. Nel nostro codice, controlliamo l'ingresso di tali sensori in ciascun ciclo della **while** e verifichiamo se uno di loro è stato attivato . Utilizziamo quindi una istruzione **if** per controllare se va effettuata una correzione della rotta oppure se si deve proseguire senza alcun aggiustamento. Quando un sensore viene attivato, il motore relativo a quel lato viene mantenuto attivo, mentre il motore opposto viene brevemente spento per ottenere di far girare il robot. Viene poi spinto un poco avanti prima di proseguire la sua strada.

Potrebbe quindi essere necessario modificare i tempi di sterzata e il successivo avanzamento al fine di ottenere che il robot segua meglio il percorso. Le temporizzazioni sono Inserite in secondi, non esistate a "giocarci" fino a ottenere il miglior compromesso possibile.

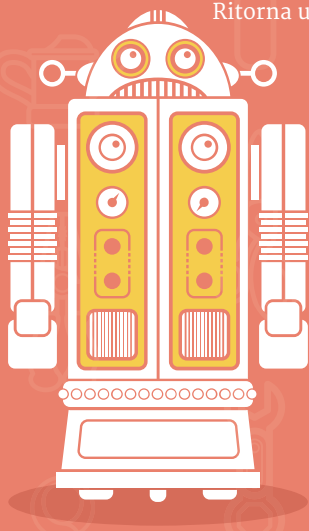
Non colpire i muri

La parte finale del codice è per l'uso del sensore ad ultrasuoni . È usato molto semplicemente per rilevare la distanza di qualsiasi oggetto che il robot possa trovarsi di fronte durante il suo percorso.
Al momento, lo abbiamo impostato In modo che se si trova a 5 cm da un oggetto, dovrebbe fermarsi così da non colpirlo. La misura **distance > 5** è relativa: a seconda del freno motore, tenuta e peso del carico sul vostro robot, dovreste poter essere costretti a modificare questo valore.

Quando il robot rileva un ostacolo sulla sua strada, si attiva una delle due parti di codice progettate per spegnerlo. Ferma i motori , fa la pulizia dei pin del GPIO (che è importante), dopodiché vi informa su quello che è successo.

Metteteci del vostro

Questo codice mostra quasi ogni funzione principale del robot . Che cosa si può fare con quanto imparato e con questo codice? Con un po' di tweaking , potreste renderlo telecomandato tramite l'invio di comandi di direzione attraverso SSH, oppure si potrebbe programmare al suo interno un percorso prestabilito da seguire.

Divertitevi con il vostro robot,e non abbiate mai paura di sperimentare e di cercare di renderlo il robot che avete sempre sognato.
Potreste anche incominciare a pensare di espandere la vostra esperienza robotica con un progetto più ambizioso e complesso, in futuro...

MagPibot.py

```
#!/usr/bin/env python

# Import libary functions we need
import RPi.GPIO as GPIO
import time
import sys
GPIO.setmode(GPIO.BCM)


# Set which GPIO pins the drive outputs are connected to
DRIVE_1 = 4
DRIVE_2 = 18


# Set the GPIO pins of the ultrasonic sensor
TRIG = 40
ECHO = 37


# Set the GPIO pins of the line sensor
CENTRE = 31
LEFT = 32
RIGHT = 33


# Set all of the drive pins as output pins
GPIO.setup(DRIVE_1, GPIO.OUT)
GPIO.setup(DRIVE_2, GPIO.OUT)


# Function to set both drives off
def RobotStop():
    GPIO.output(DRIVE_1, GPIO.LOW)
    GPIO.output(DRIVE_2, GPIO.LOW)


# Function to set both drives on
def RobotForward():
    GPIO.output(DRIVE_1, GPIO.HIGH)
    GPIO.output(DRIVE_2, GPIO.HIGH)


# Set ultrasonic pins function
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)


# Give sensor time to settle
GPIO.output(TRIG, False)
print “Waiting for ultrasonic sensor to settle...”
time.sleep(2)
print “Settled!”


# Set line follower pins as inputs
GPIO.setup(CENTRE, GPIO.IN)
GPIO.setup(LEFT, GPIO.IN)
GPIO.setup(RIGHT, GPIO.IN)


try:
    # Start by turning all drives off
    RobotStop()
    raw_input(“We’re ready to roll, press ENTER to continue”)
    while True:
        # Get distance to object

        GPIO.output(TRIG, True)
        time.sleep(0.00001)
        GPIO.output(TRIG, False)


        while GPIO.input(ECHO)==0:
            pulse_start = time.time()


        while GPIO.input(ECHO)==1:
            pulse_end = time.time()


        pulse_duration = pulse_end - pulse_start


        distance = pulse_duration * 17150


        distance = round(distance, 2)


        # Check line followers
        line_left = GPIO.input(LEFT)
        line_right = GPIO.input(RIGHT)


        if distance > 5:
            if line_left == 0:
                # Drive 1 state
                GPIO.output(DRIVE_1, GPIO.HIGH)
                # Drive 2 state
                GPIO.output(DRIVE_2, GPIO.LOW)
                # Get back on course
                time.sleep(1)
                RobotForward()
                time.sleep(0.5)
            elif line_right == 0:
                # Drive 1 state
                GPIO.output(DRIVE_1, GPIO.LOW)
                # Drive 2 state
                GPIO.output(DRIVE_2, GPIO.HIGH)
                # Get back on course
                time.sleep(1)
                RobotForward()
                time.sleep(0.5)
            else:
                # Move forward
                RobotForward()
        else:
            RobotStop()
        print ‘I have stopped, please turn off my power!’
            GPIO.cleanup()
        sys.exit()


except KeyboardInterrupt:


        # CTRL+C exit, turn off the drives and release GPIO pins
        print ‘Stopped early, please turn off my power!’
        RobotStop()
        GPIO.cleanup()
```